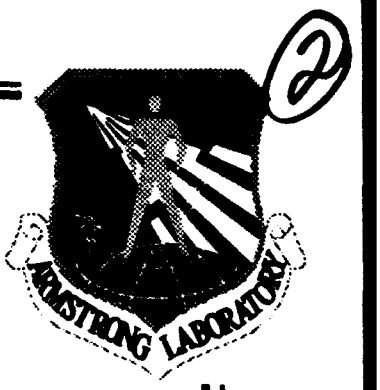
AL/HR-TR-1993-0132



APPLICATION OF DIGITAL VIDEO TO HYPERMEDIA TRAINING SYSTEMS

Lee Lacy
Richard Braby
De'Pass Smith

Logicon Eagle Technology, Incorporated
950 N. Orlando Ave., Suite 200
Winter Park, FL 32789

Mike Slaven, Captain, USAF
Kevin Glass, Captain, USAF

HUMAN RESOURCES DIRECTORATE
TECHNICAL TRAINING RESEARCH DIVISION
INTELLIGENT TRAINING BRANCH
7909 Lindbergh Drive
Brooks Air Force Base, TX 78235-5352

November 1993

Final Technical Report for Period January 1991 – March 1993

AIR FORCE MATERIEL COMMAND
BROOKS AIR FORCE BASE, TEXAS

NOTICES

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely Government-related procurement, the United States Government incurs no responsibility or any obligation whatsoever. The fact that the Government may have formulated or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication, or otherwise in any manner construed, as licensing the holder, or any other person or corporation; or as conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

The Office of Public Affairs has reviewed this report, and it is releasable to the National Technical Information Service, where it will be available to the general public, including foreign nationals.

This report has been reviewed and is approved for publication.

Kurt W. Steuck

KURT W. STEUCK, Ph.D.
Contract Monitor

James W. Parlett

JAMES W. PARLETT, Lt Col, USAF
Chief, Intelligent Training Branch

Rodger D. Ballentine

RODGER D. BALLENTINE, Colonel, USAF
Chief, Technical Training Research Division

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	November 1993	Final – January 1991 – March 1993

4. TITLE AND SUBTITLE

Application of Digital Video to Hypermedia Training Systems

5. FUNDING NUMBERS

C – F33615-91-C-0003
PE – 62205F
PR – 1121
TA – 09
WU – 80

6. AUTHOR(S)

Lee Lacy
Richard Braby
De'Pass Smith
Mike Slaven
Kevin Glass

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Logicon Eagle Technology, Incorporated
950 N. Orlando Ave., Suite 200
Winter Park, FL 32789

8. PERFORMING ORGANIZATION REPORT NUMBER

879.01.4

9. SPONSORING/MONITORING AGENCY NAMES(S) AND ADDRESS(ES)

Armstrong Laboratory (AFMC)
Human Resources Directorate
Technical Training Research Division
7909 Lindbergh Drive
Brooks Air Force Base, TX 78235-5352

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

AL/HR-TR-1993-0132

11. SUPPLEMENTARY NOTES

Armstrong Laboratory Technical Monitor: Dr. Kurt W. Steuck, (210) 536-2034

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

A research effort was conducted to analyze the potential for using autoauthoring training hypermedia in teaching students to perform procedures. In this effort, autoauthoring involved automating the creation of training material from a description of a procedure rather than using traditional methods such as flowcharting and programming. The Prototype Animated Training Hypermedia System (PATHS) was the computer program developed to perform this process. Sample applications were produced using PATHS, demonstrating the feasibility of this concept for procedural trianing. PATHS rapidly generates training hypermedia. A prototype, PATHS, needs further testing and development before it can become an operational system.

14. SUBJECT TERMS

Autoauthoring
Digital video
DVI
Hypermedia
Instructional development

15. NUMBER OF PAGES

38

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

Table of Contents

List of Figures

PREFACE

This effort investigated methods of producing training hypermedia that could be more cost-effective than traditional methods. It is one component of a research program being conducted by the U.S. Air Force's Armstrong Laboratory into the use of advanced computer technology to create and deliver technical instruction.

SUMMARY

A research effort was conducted to analyze the potential for using autoauthoring training hypermedia in teaching students to perform procedures. In this effort, autoauthoring involved automating the creation of training material from a description of a procedure rather than using traditional methods such as flowcharting and programming. The Prototype Animated Training Hypermedia System (PATHS) was the computer program developed to perform this process. Sample applications were produced using PATHS, demonstrating the feasibility of this concept for procedural training. PATHS rapidly generates training hypermedia. A prototype, PATHS needs further testing and development before it can become an operational system.

APPLICATION OF DIGITAL VIDEO TO HYPERMEDIA TRAINING SYSTEMS

I. INTRODUCTION

Objective

The objective of this project was to investigate the application of digital video in the design and delivery of hypermedia. The objective of the hypermedia content was to teach students to perform procedures and the process of creating training hypermedia seemed a likely candidate for autoauthoring. Autoauthoring automatically produces a full set of hypermedia screens from procedural data stored in a database. The contract specifications required formatting the screens with algorithms to produce demonstrations, simulations, scored exercises and tests, and job performance aids incorporating valid learning principles. By eliminating labor intensive steps in designing training hypermedia through the use of autoauthoring, development costs were expected to be substantially reduced.

Background

Traditional methods of developing interactive courseware involve extensive preparation of flowcharts and storyboards to organize information in the training application design. (Bunzel, 1992) This development strategy results in a stack of paper documents describing technical content to be taught and instructional treatment. The design is then implemented by programming lessons using an authoring system. Often, the same data is entered at various locations as part of different instructional "treatments". When changes to a procedure occur, these lessons become difficult to maintain because each element of information to be updated, such as a step in the procedure, is repeated many times throughout the

various demonstrations, exercises, tests, and job aids. Each instance must be separately edited.

The author's task becomes especially difficult when the requirement is to develop hypermedia with extensive linking and cross-referencing. Hypermedia is the combination of hypertext and multimedia and includes the ability to navigate through a linked multimedia document. Hypertext refers to a computer-based document composed of text and graphics which contains links making it possible for a user to navigate through the document in a non-linear fashion. Multimedia refers to the presentation of a variety of media including text, still graphics, video, audio, and animation.

The purpose of all hypermedia is to convey information, but a subset of these documents is tailored toward training. Training hypermedia refers to hyperdocuments (hypermedia documents) which are designed to be used in technical training programs. Quality training hypermedia must contain the types of presentations needed for training including instructional support features which assist the user in learning the material.

Training hypermedia offers several advantages over traditional computer-based instruction. By including additional media, such as video and animation, students find the material realistic and more interesting to use. By supporting learner control (the ability for the student to determine a path through the subject matter) students experience increased motivation. (Jonassen, 1988)

However, problems have been documented concerning the development and use of training hypermedia. A primary problem is that hypermedia-based instruction is difficult to develop. A great deal of time must be spent in media preparation and in programming the links that connect individual nodes in the hyperdocument. Many hypermedia systems begin as rapid prototypes. It is often difficult to develop a usable

system from a prototype stage product. (Glushko, 1992) Other problems relate to the use of training hypermedia. Users of hypermedia often get lost in "hyperspace". They are unable to determine their location in the hyperdocument, and they are uncertain which node to visit next. (Marchionini, 1988)

If an efficient form of hypermedia could be developed to support students in learning to perform procedures, it would spawn many spinoff applications because a major component of technical training deals with teaching students to perform procedures; more efficient ways of achieving this type of learning are always of interest to training system designers. Applications are easily identified. For instance, various medical practitioners must learn operating procedures for medical equipment. In addition to the general task of learning to operate equipment, most equipment maintenance actions have been proceduralized.

Procedures which are candidates for use in autoauthored instruction are characterized by discrete steps performed on a system. This proceduralized process can be easily represented in a computer database. Experience has shown that this type of database is best authored by an expert on the procedure, rather than by a programmer. (Clark, 1991)

The concept tested in this project was whether training material could be generated using a database containing procedural information and the required formatting algorithms. It was further proposed that each of these training formats could be designed to incorporate features that support efficient learning. For instance, simple simulations could be designed to provide students with an opportunity to practice performing a procedure. Often a "cause and effect" type of simulation is sufficient for conveying a concept. (Mears, 1989) An object, such as a switch, can have an associated initial view which changes to a resulting view when an action is performed on the object. These simulations should also be capable of presenting secondary effects (responses) as a result of the desired action. For example, when

a student performs a "power up" step, an "off" view of a toggle switch could be changed to an "on" view when the student clicks on the switch object. Then the "power on light" would be illuminated as the response.

The hypothesis was that by using computer routines to create a database on procedures performed on a system, and then generating hypermedia using the autoauthoring routines, first draft training hyperdocuments could be generated. These draft hyperdocuments could be used to replace the output of traditional flow charts and storyboards in the review process of developing interactive courseware. By reviewing and editing the database and rapid prototypes, final hypermedia could be efficiently generated.

Organization of Report

This report describes the development and demonstration of the Prototype Animated Training Hypermedia System (PATHS), a prototype autoauthoring system, with the characteristics outlined in the preceding paragraphs. This final report is divided into five major sections. In addition to this Introduction, other sections included in this final report are the Technical Approach, Results, Conclusions and Recommendations. The Technical Approach section describes the activities conducted as part of this effort. The Results section describes the hardware, software, and documentation resulting from this effort. The Conclusions section lists major insights gained during the process of developing and initially testing the system. And the Recommendations section lists suggestions for the future application and improvement of PATHS.

II. TECHNICAL APPROACH

This effort included research, rapid prototyping, design, coding, application development, and testing of PATHS. Research was conducted to define the goals of the effort. The requirements were further refined by rapid prototyping. PATHS was then designed and coded using lessons learned from the rapid prototyping phase. Sample applications were developed using the system. The system was tested and the autoauthoring routines were iteratively refined during the development of the sample applications, and by repeated trials, the various computer routines.

Research

A literature search was conducted to investigate related efforts and to obtain information concerning training hypermedia systems. Subjects of investigation included training hypermedia, instructional development, and digital video technology. Many of the books and articles reviewed are listed in the Bibliography.

A related effort which significantly impacted the design of PATHS was identified during the research phase. A tri-service initiative called the Interactive Electronic Technical Manual (IETM) effort is involved in defining standards for the interactive presentation of technical manual information for operational and training purposes.

The research phase helped refine the requirements of the system. For example, the research literature on browsers influenced the browser user interface and type of hypermedia to be used. Also, the need for a "back on track" icon was identified which guides the student's return to a predefined path. The refined requirements were documented in a Software Requirements Specification (SRS).

Rapid Prototyping

Rapid prototypes of the PATHS hardware/software configuration were developed to determine the useability of digital video using the Intel ActionMedia I board. Rapid prototypes are temporary configurations created outside the traditional development process in order to quickly identify potential problem areas and investigate solutions. Although Intel's Digital Video Interactive (DVI®) technology had been selected to support digital video presentation requirements, it was not clear how extensively DVI could be used for user-interface support. As a result of rapid prototyping, it was decided that DVI graphics would be used for the browser user interfaces, and that video would be shown in full-screen mode to avoid lip-syncing problems.

Design

The goal of the design phase was to plan the implementation of PATHS based on the requirements described in the SRS. A heavy emphasis was placed on providing complete functionality and documenting the approach in a detailed design document prior to the beginning of software coding.

A design was developed which incorporated a concept similar to "view packages" in interactive electronic technical manual (IETM) systems (See Figure 1). An instructional developer created the conceptual structure of a database, referred to as the Application Knowledge Base (AKB). This database structure was then used to create the autoauthoring routines used to generate a hyperdocument. A design goal was to make the hyperdocument generic enough so that a commercial off-the-shelf (COTS) browser could eventually be used to view and modify the hyperdocument. Because the U.S. Air Force (USAF) did not want to select a target COTS browser at this stage in the research, a custom hypermedia editor browser was designed for

PATHS. The problem in making the hypermedia browser generic is that the browser cannot perform intelligently based on the type of data (procedural training) that it is manipulating.

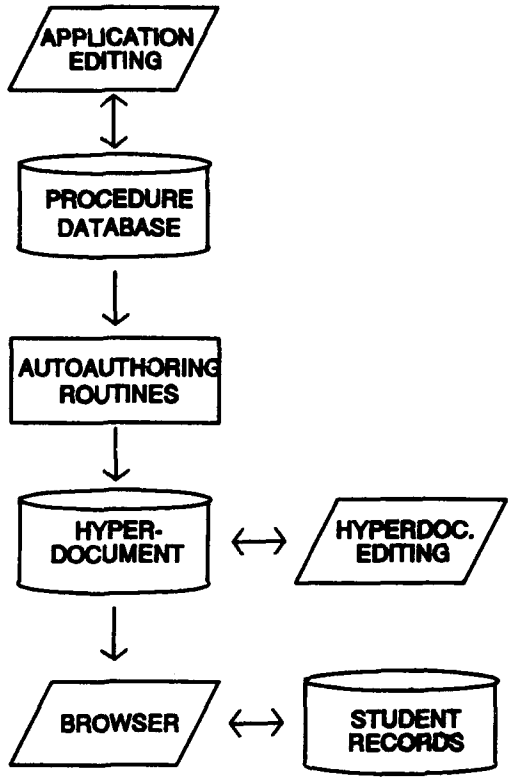


Figure 1. - PATHS Concept

A basic concept supported in this design phase was that data stored in the AKB could conceivably be imported from external sources. Similarly, the hyperdocument might be exported to other browsers. Therefore, a standard file format was desired to represent both databases. The Standard Generalized Markup Language (SGML) was selected for representing these databases. An international standard for defining hypermedia DTDs, the Hypermedia/Time-based (HyTime) standard was used to define the Data Type Definitions (DTDs) for the databases. The PATHS DTDs were developed with the assistance of the HyTime author, Dr. Steven Newcomb. The PATHS DTDs are presented in the PATHS Software Detailed Design (SDD). Although PATHS is not fully SGML-compliant, the data is tagged and stored in an ASCII file.

Once the design overview and file representations were determined, a detailed design was conducted. The design centered around providing the functionality of the four processes: AKB editing, autoauthoring, hypermedia editing, and browsing.

The AKB editing functionality was to allow users to create and modify the contents of the AKB. Therefore, the design of the AKB editing routines was centered around developing a minimal set of screens to allow the user to easily enter all the

required data without redundancy. Descriptions of the AKB editor screens are presented in the SDD.

The design of the PATHS autoauthoring routines are based on instructional strategy algorithms developed for this effort. These algorithms establish standard formats for the procedure demonstration, simulation, performance tests, and other procedural presentations. They also control what elements of information are pulled from the AKB for use in filling in the formats. They also structure the student's interaction with these displays. Flowcharts which represent these algorithms are presented in the SDD.

The AKB is populated using the AKB editor prior to autoauthoring. However, there may be occasions when the standard text built into the formats by the autoauthoring routines must be edited for specific applications. The design of the hypermedia editor was similar to the AKB editor. For both editors, screen designs were the primary focus. The screens for the hypermedia editing portion of the system are presented in the SDD.

The browser design effort was focused on presenting generic hypermedia material in a non-frame-based hypermedia browser with traditional navigation capabilities. The generic hypermedia material to be presented included text, still graphics, audio, and video. The navigation capabilities included linking through text buttons, hot spots on graphic stills, and icons.

Based on the data elements in the DTDs and the user interface elements in the screen designs, an object-oriented design was developed. There are four major groups of objects in the design:

- User Interface objects
- Application Knowledge Base objects

- Hypermedia primitive objects
- Student Records objects

The user interface objects include items such as menus, forms, fields, and windows. These objects were designed to map closely to the COTS user interface support libraries selected for this effort.

The application knowledge base objects were designed to match the structure of knowledge used to describe the AKB (See figure 2).

The hypermedia primitive objects were intended to represent the primitive data types used for presentation in the browser. The primitive data classes are tied closely to support from the Intel ActionMedia software libraries. These libraries provide low-level support for the Intel ActionMedia I (DVI). The DVI board provides not only full-screen full-motion digital video, but provides graphics primitives.

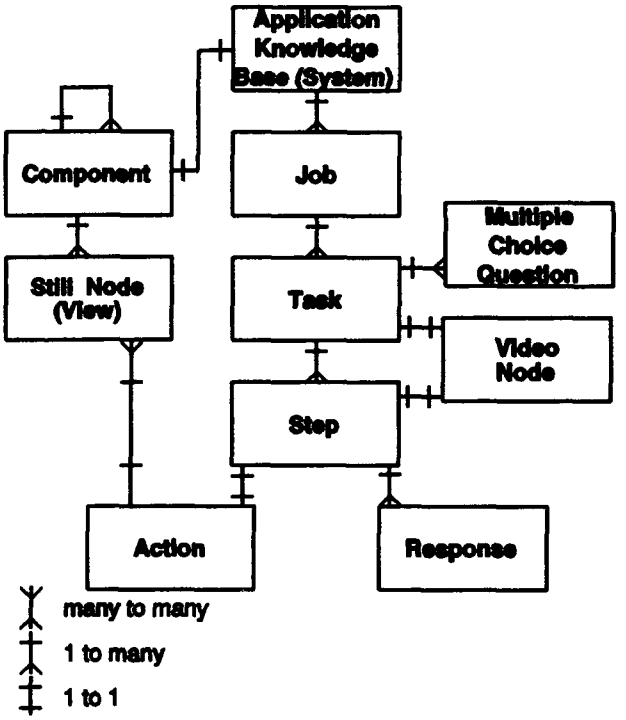


Figure 2. - Application Knowledge Base Structure

Student records objects are used to manipulate the student registration and student session data.

The detailed results of the design process, updated to reflect the "as-built" product, are presented in the Software Design Document (SDD).

Coding

Once the design was approved at an informal Critical Design Review (CDR), the coding phase began. The initial version of the software was generated by translating Program Design Language (PDL) into "C" statements. Detailed PDL was included in the preliminary version of the SDD. The detailed PDL was easily translated and the conversion process was accelerated by converting word processing documents containing PDL to ASCII files which eventually contained "C" code.

PATHS was developed using Microsoft® "C" version 5.1. Commercial off-the-shelf software libraries were used to avoid writing low-level code for user interface support. The libraries used were Vermont Views™ version 2.0 and the ActionMedia™ 750 (DVI) Software Libraries. The project was originally intended to be written in "C++". However, the DVI libraries could only be linked to Microsoft "C" compiled code, and were incompatible with Microsoft's "C++" (Version 7) compiler when it became available.

During the creation of early versions of PATHS, the size of the executable file resulting from the linking process continued to grow, and it became clear that advanced memory management strategies must be employed. The first attempt at freeing memory was performed by optimizing the environment. DOS 5.0, which has a small kernel, was installed. The 386MAX®6 utility was also installed to move some of the device drivers to high memory; it freed a considerable amount of memory, but not enough. An overlay linker, .RTLink®Plus, was employed to swap code in and out of memory. However, both the Vermont Views software library routines and the ActionMedia Software Library routines would not function properly unless they were allowed to remain in memory. The largest portion of memory was used by these libraries; because the benefit of the overlay linker was negligible, it was eventually discarded.

The application was broken into five separate executables in order to overcome memory constraints. None of the five required both the Vermont Views libraries and the ActionMedia Software Libraries. The functionality of the five executables was selected based on the modularity of the design. Many of the software modules are shared between the five executables. A matrix describing the reuse of the modules is presented in the SDD.

As code was generated, test data sets were created to test the executables. All PATHS data is tagged ASCII data, which makes it easier to debug the data entry, autoauthoring, and browsing features because the data can be easily viewed in an editor. A design goal was to allow a developer to partially specify a procedure, and then review the resulting hyperdocument. The lack of data in these test data sets called attention to the need for and development of graceful degradation features. The code was iteratively refined as features were added and test data sets were modified and used.

The detailed results of the coding effort are documented in the Training User's manual and the Software Product Specification (SPS). The Training User's Manual provides the steps for using each of the five executables. The SPS contains the SDD and code listings. The code listings are generated from the baseline version of the software. The baseline is maintained using the Polytron Version Control System (PVCS). Using PVCS, each modification to baseline code can be tracked.

Problems in coding resulted primarily from DOS limitations and the ActionMedia 750 Software Libraries. New support software is now available for DVI in the Microsoft Windows™ environment. The memory management features of Windows allow simpler access to all system memory, and they are more advanced than DOS. These features should enable developers to create much larger DVI-based applications than they could in DOS. Also, new standards for interacting with digital video under Windows, such as Video for Windows, are emerging. The inability to link "C++"

code to the ActionMedia libraries was very unfortunate because the advantages of "C++" would have benefited the elegance of the design and the reusability of the code. The use of "C++" would be possible for a system operating under Windows 3.1+ and Video for Windows.

Application Development

The purpose of the PATHS software is to allow instructional developers to develop training hypermedia applications which can be used by students. Applications are developed using PATHS' five executable routines. Detailed instructions for using these routines can be found in the PATHS User's Manual.

Using PATHS, the emphasis in instructional development is on media preparation and data management instead of design and programming. The media must be properly located on the hard disk for the autoauthored hyperdocument to access the media for presentation. Although applications can be developed in a number of ways, a preferred method is described below.

First, the steps of a procedure are entered by filling form screens in the AKB editor. References to associated media are also entered. The media must be captured, converted to the required format, and stored in the proper locations on the hard disk. Procedures are made up of individual steps, which may have associated responses.

Graphic stills associated with the steps can be generated in a number of ways. In this effort, stills were created by using a color scanner to digitize photographs of the subject material. A translation process was used to convert the digitized photographs to the desired size and format for the hypermedia browser.

Video clips can be generated in two ways. The DVI capture board can be used to create a low quality compression. Alternatively, a high quality compression facility can be used to create high quality digital files from the video. Most of the video clips created for this effort were digitized directly from an output of a professional video camera. Others were made by digitizing from videodisc and a VHS video tape.

A demonstration video was prepared as part of the task of documenting the results of this effort. The 5-minute video describes the overall concepts of the PATHS system and shows sample screens from each of the five PATHS executable programs.

Testing

Several types of software testing were conducted including graceful degradation tests, and tests to verify the completeness of sample applications, and correct implementation of the design.

Graceful degradation is desirable in order to create review hyperdocuments from partially specified procedures and media. Sample applications assisted the testing process by providing test data. Analyses were made of the PATHS hyperdocuments to ensure that they conformed to the instructional strategy described in the design information.

III. RESULTS

PATHS Hardware and Software

Off-the-shelf hardware components were assembled into a PC-based multimedia workstation. COTS software packages were installed on the hardware. This hardware/software configuration was used to develop and host custom software generated to provide the PATHS functionality.

Five executable programs were created and demonstrated. They are:

- PATHS AKBEDIT, a data base editor
- PATHS AUTOAUTH, an autoauthoring function
- PATHS TOCIT, a table of contents generator
- PATHS HYPERED, an editor of hypermedia screens
- PATHS BROWSE, the student's interface with hyperdocuments

These executable programs are compiled from approximately 15,000 source lines of code. All of these programs are functional, although memory problems limit the size of the applications that can be generated and used.

The PATHS AKBEDIT program is used to populate the AKB with text and graphic data which describe a procedure, and the equipment on which the procedure is performed. AKBEDIT is also used in editing the contents of the AKB, once it is generated. Data on a procedure is entered with limited regard to how or where it will be used in the various training applications of the data.

The PATHS AUTOAUTH software is used to autoauthor the hyperdocument, including a sequence of displays for procedure demonstrations, practice exercises, tests and job performance aids. This function replaces the labor intensive task of programming with an authoring system to create a hyperdocument. As an example of its speed, over a thousand screens of a hyperdocument, including all links, can be reliably generated in less than a minute.

The PATHS TOCIT program is used to create a table of contents map of the hyperdocument so that students can determine their location and move around within the hyperdocument. This textual table of contents provides a detailed picture of the hyperdocument structure.

The PATHS HYPERED software is used to edit the hyperdocument created by AUTOAUTH. It is used to respond to unique situations and customize applications and should be used only when necessary; the changes made with this program do not survive when the AUTOAUTH program is repeated. Its purpose is to allow changes to be made to a final version of a hyperdocument before it is used by students.

The PATHS BROWSE program is used to present a hyperdocument to the student, and to allow the student to interact with the hyperdocument. The primary screen from this program is shown in Figure 3.

Sample Lessons

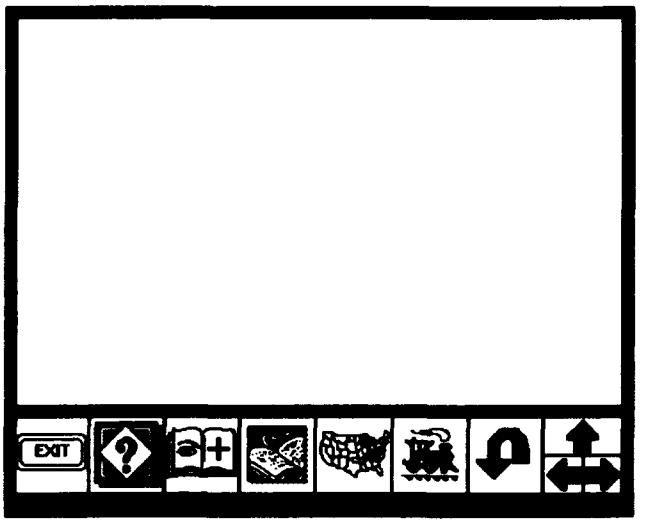


Figure 3. - BROWSE Screen

Two sample hyperdocuments were created to demonstrate the PATHS hardware and software.

The first sample application presented a procedure performed on the MiniOxIII, an oxygen analyzer used by medical personnel to monitor the flow of oxygen being given to patients being evacuated by aircraft. Altitude changes make it necessary to closely monitor this function. The particular procedure presented is one for calibrating the MiniOxIII at the homestation. This procedure is typical of a general procedure performed on electronic equipment.

The second procedure concerns the use of a mechanical bracket used to hold one corner of a stretcher during an air evacuation mission. It is a six-step procedure performed on a piece of mechanical equipment, typical of many mechanical operations performed on small pieces of equipment.

In the development and testing of these two applications, all system functions were evaluated.

Documentation

Documentation was generated to guide USAF personnel in using and maintaining PATHS. A User's Manual was created containing detailed directions for using PATHS programs to produce and use hyperdocuments. A short videotape was

produced to introduce the PATHS system to new users. PATHS has been described in papers presented at professional conferences and published in the conference proceedings. Articles appear as:

"Autoauthoring Medical Training Hypermedia", *Proceedings of the 11th Annual Conference on Interactive Instruction Delivery*, (SALT '93), Orlando, Florida, February 24-26, 1993.

and

"Autoauthoring Procedural Training Hypermedia", *Proceedings of the Interservice/Industry Training Systems and Education Conference*, (I/ITSEC '92), San Antonio, Texas, November 2-4, 1992.

Observations were made on the significance of this effort, and how to approach the further development of the autoauthoring concept.

The software development effort was documented to MIL-STD-2167A/T specifications. Documentation included:

- Software Requirements Specification (SRS)
- Software Detailed Design (SDD)
- Software Product Specification (SPS)

IV. CONCLUSIONS

There were several conclusions derived from this research effort. They are presented in the following sections, along with supporting information.

Conclusion #1: Autoauthoring techniques for creating hyperdocuments to teach procedures have been proven technically feasible.

- Procedures are an acceptable domain for autoauthoring

  Procedures (tasks) are made up of individual steps composed of actions and responses. The inherent structure of procedures makes it possible to create a database containing the procedure's elements; the database can provide these data elements to construct algorithms.

- Instructional design algorithms are codeable

  The algorithmic production of hypermedia from data in a database utilizes simple compiler technology and does not require artificial intelligence techniques such as neural networks or expert systems rule-based systems.

- User-friendly hypermedia can be generated

  Often, the output of computer processes is terse and presents too much information on a single screen. However, computer programmers can design user-friendly training screens that are much less threatening to the student.

- Meaningful instructional sequence can be generated

  Algorithm-driven sequencing of instruction can determine what information is available, and which screens should precede or follow others.

- Graceful degradation can be utilized

  During early stages of development, some information may be missing. Autoauthoring routines can be created to generate as much material as possible (given the existing data at the time of autoauthoring) that do not require a set amount or type of data to generate a hyperdocument.

Conclusion #2: Autoauthoring techniques can substantially reduce the cost of producing procedure-oriented interactive courseware.

- Traditional methods require substantial investment

  Hand-crafting training hypermedia involves the use of flowcharts, storyboarding, and programming. These methods often take hundreds of hours and thousands of dollars to create an hour of quality instruction.

- Autoauthoring can rapidly generate consistent hypermedia

  PATHS can generate over a thousand frames of hypermedia in less than a minute. The created hyperdocument does not require traditional testing for branching and sequencing because the material is computer-generated.

- Emphasis shifts to data management from programming

  Although the coding portion of PATHS instructional development is automated, information is still required to perform the automation. Procedures must be entered, and media must be prepared.

- Algorithms define the learning strategy

  Traditional design work is eliminated because PATHS contains learning strategies which do not have to be redesigned for each domain.

Conclusion #3: User trials are needed for further development and new hardware/ software is needed to support these tests.

- There is a new generation of DVI boards

  The current PATHS configuration uses ActionMedia I boards; ActionMedia II boards generate improved digital video.

- New memory management is available through Windows

  PATHS is implemented in five separate executables due to memory limitations of DOS. By migrating the software to the Windows environment, the system can be combined into a single application and can take advantage of digital video standards such as Video for Windows.

- User trials should be performed to tune the system

  PATHS should be used by instructional developers, students, and instructors to determine changes required to meet the needs of each user group. No such tests were conducted as part of this effort. These trials should be performed informally on the existing version of PATHS before the next version is developed.

Conclusion #4: Generic Browsers which present non-frame-based hypermedia from a compiled format should be used to present training.

- Generic hypermedia can be exported to COTS browsers

  The design PATHS hyperdocuments was kept as generic as possible to allow for future use of COTS hypermedia browsers. A COTS browser reduces cost, and has greater availability and more advanced features.

- Non-frame-based hypermedia is superior to frame-based systems

  Hypermedia should be created using a non-frame-based approach. When media is inserted into the frame of a frame-based hyperdocument, the frames may require editing. Non-frame-based hypermedia is formatted on-the-fly and does not require modification.

- Results of autoauthoring process should be modifiable

  By compiling the instructional material to a "view package", modifications can be made to the generated material, rather than having to accept the computer's choices for every screen.

Conclusion #5: Developing Autoauthoring software presents unique challenges.

- A flowcharting tool could improve the development process

  A Computer Aided Software Engineering (CASE) flowcharting tool would greatly simplify descriptions of the instructional flow which makes up the hypermedia web. Automating this process would also simplify the process of verifying connectivity between flowcharts.

- The use of an instructional strategist is critical

  It is important to employ an expert at instructional strategy when developing and coding instructional sequence and learning strategies. All

of the applications generated will reflect the design strategies programmed into the autoauthoring routines.

- Developers should keep up with "bleeding edge" technology

  Multimedia and hypermedia technology continues to develop rapidly, and solutions are becoming available to many of the deficiencies of current technology. By investing in evolving baselines, the resulting product will remain current and marketable at the conclusion of the development effort.

Conclusion #6: DoD Autoauthoring tools should import IETM data.

- Importing data reduces costs

  New DoD standards have been developed for describing the format of electronic technical manual data (including procedures). This data should be input into autoauthoring of training materials to reduce the data entry requirement and create consistency between operational and training uses of the data. This methodology can lower costs arising from reduced data-entry verification and maintenance requirements when system changes occur.

- COTS Software Components support SGML processing

  COTS software libraries are becoming available for importing and exporting CALS SGML-tagged data into applications. The use of these libraries can reduce the cost of system development by reducing the amount of custom code required.

V. RECOMMENDATIONS

Based on the results and conclusions of this effort, several recommendations are listed below.

Recommendation #1. An informal evaluation of PATHS hyperdocuments should be conducted.

Students and instructors should be asked to comment on the usefulness of sample hyperdocuments, and suggest improvements. The current version of PATHS should be used for this evaluation.

Recommendation #2. Develop a second generation PATHS

The lessons learned from this effort and new related technology should be integrated in a second generation system. This system should include an IETM interface to input procedures, and it should create hyperdocuments that can be read by COTS browsers. The systems should use the new DVI II boards and Video for Windows, operating under Windows 3.1+. It should be written in "C++" using Object-oriented Design principles.

Recommendation #3. Conduct a formal field test of the second generation PATHS.

A practical set of hyperdocuments should be developed using the new system. The instructional developer's process using the new system should be compared to traditional methods. Similarly, the performance of students using PATHS hyperdocuments should be compared with the performance of students using traditional methods. The author's productivity in terms of resources (time/funds) required to produce hyperdocuments with PATHS and with traditional hyperdocument authoring systems should be analyzed.

REFERENCES

Bunzel, M. & Morris, S. (1992). *Multimedia Applications Development using DVI Technology*, New York: McGraw Hill.

Clark, D. R. (1991, July). "Demise of Multimedia", *IEEE Computer Graphics & Applications*, IEEE. pp. 75-80.

Glushko, R. J. (1992, 2Qtr). "Seven Ways to Make a Hypertext Project Fail", *Technical Communications*. pp. 226-230.

Jonassen, D. H. (1988, November). "Designing Structured Hypertext and Structuring Access to Hypertext", *Educational Technology*. pp. 13-16.

Jonassen, D. H. (1986). "Hypertext Principles for Text and Courseware Design", *Educational Psychologist*. pp. 269-292.

Marchionini, G. (1988, November). "Hypermedia and Learning: Freedom and Chaos", *Educational Technology*. pp. 8-12.

Mears, J. E., Braby, R., Hebert, J., Babu, A. J. G., Kincaid, J. P., & Meliza, L. L. (1989, October). "Organization of Technical Information for Electronic Delivery", *ARI Research Report 1543*. U.S. Army Research Institute for the Behavioral and Social Sciences.

BIBLIOGRAPHY

Akscyn, R. M., McCracken, D. L., & Yoder, Elise A. (1988). "KMS: A Distributed Hypermedia System for Managing Knowledge in Organizations". *Communications of the ACM*, 31, 820-835.

Ambron, S. & Hooper, K. (Eds.). (1990). *Learning with Interactive Multimedia*. Redmond WA: Microsoft Press.

Beck, J. R., & Spicer, D. Z., (1988, February). "Hypermedia in Academia". *Academic Computing*, pp. 22-50.

Bryan, M. (1988). *SGML an Author's Guide*. Addison-Wesley.

Bush, V. (1945). "As We May Think". *Atlantic Monthly*, pp. 101-108.

Campbell, B. , & Goodman, J. M. (1988). "HAM: A General Purpose Hypertext Abstract Machine". *Communications of the ACM*, 31, 856-861.

Carlson, P. A. (1989). "Hypertext and Intelligent Interfaces for Text Retrieval". In *The Society of Text*. Cambridge, MA: MIT Press.

Carlson, P. A. (1989). "Hypertext: A Way of Incorporating User Feedback into Online Documentation". In *The Society of Text*. Cambridge, MA: MIT Press.

Conklin, J. (1987, September). "Hypertext: An Introduction and Survey". *IEEE Computer*, pp. 17-41.

Dixon, D. F. (1989). "Life Before the Chips: Simulating Digital Video Interactive Technology". *Communications of the ACM*, 32, 824-831.

Dixon, D. F. (1987, July). "DVI Video/Graphics". *Computer Graphics World*. pp. 125-128.

England, N. (1986, January). "Graphics System Architecture for Interactive Application-Specific Display Functions". *IEEE CG&A*. pp. 60-70.

Fiderio, J. (1988, October). "A Grand Vision". *Byte*. pp. 237-244.

Fox, E. A. (1989). "The Coming Revolution in Interactive Digital Video". *Communications of the ACM*. 32, 794-801.

Frenkel, K. A. (1989). "The Next Generation of Interactive Technologies", 7/89, *Communications of the ACM*, 32, 872-881.

Frisse, M. E. (1988, October). "From Text to Hypertext". *Byte*. pp. 247-253.

Frisse, M. E. (1988). "Searching for Information in a Hypertext Medical Handbook". *Communications of the ACM*. 31, 880-886.

Garg, P. K. (1988). "Abstraction Mechanisms in Hypertext". *Communications of the ACM*. 31, 862-879.

Gay, G. & Raffensperger, E. (1989, September). "Considerations and Strategies in the Design of Interactive Multimedia Programs". *Academic Computing*. pp. 33-36.

Halasz, F. G. (1988). "Reflections on Notecards: Seven Issues for the Next Generation of Hypermedia Systems", *Communications of the ACM*. 31, 836-852.

Herrstrom, D. S. & Massey, D. G. (1989). "Hypertext in Context" In *The Society of Text*. pp. 45-58. Cambridge, MA: MIT Press.

Hodges, M. E., (1989). "Investigations in Multimedia Design Documentation". In *The Society of Text*. pp. 79-89. Cambridge, MA: MIT Press.

Horn, R. (1989). *Mapping Hypertext*, Lexington, MA: Lexington Institute.

ISO/IEC DIS 10744.1.1, "Information Technology - Hypermedia/Time-based Structuring Language (HyTime)".

Jonassen, D. H., (1989). *Hypertext/Hypermedia*. Educational Technology Publications.

Jonassen, D. H., (1988, December). "Thinking Technology". *Educational Technology*. pp. 33-34.

Kraft, R. N. (1987). "The Influence of Camera Angle on Comprehension and Retention of Pictorial Events". *Memory & Cognition*. 15(4), 291-307.

Lippman, A. & Butera, W. (1989). "Coding Image Sequences for Interactive Retrieval", *Communications of the ACM*. 32(7), 852-859.

Loveria, G. & Kinstler, D. (1990, Fall). "Multimedia: DVI Arrives". *Byte IBM Special Edition*. pp. 105-108.

Luther, A. C. (1989). *Digital Video in the PC Environment*, McGraw-Hill.

Mackay, W. E. & Davenport, G. (1989). "Virtual Video Editing in Interactive Multimedia Applications". *Communications of the ACM*. 32(7), 802-810.

Martin, J. (1990). *Hyperdocuments & How to Create Them*, Prentice-Hall.

McBride, W. G. Jr. & Brown, S. F. (1990). "Using DVI Technology in an Intelligent Tutoring System for Artillery Forward Observers", *Proceedings of the Eighth Annual Conference on Technology and Innovations in Training and Education (TITE '90)*. (pp. 259-273).

Merrill, P. F. (1978). "Hierarchical & Information Processing Task Analysis: A Comparison". *Journal of Instructional Development*. 35-41.

Raymond, D. R. & Tompa, F. W. (1988). "Hypertext and the Oxford English Dictionary". *Communications of the ACM*. 31(7), 871-879.

Ripley, G. D. (1989). "DVI - A Digital Multimedia Technology". *Communications of the ACM*. 32(7), 811-822.

Rubens, P. (1989). "Online Information, Hypermedia, and the Idea of Literacy". In *The Society of Text*. pp. 3-20. Cambridge, MA: MIT Press.

Seyer, P. (1991). *Understanding Hypertext Concepts and Applications*. Windcrest/McGraw-Hill.

Shneiderman, B. (1989). "Reflections on Authoring, Editing, and Managing Hypertext". In *The Society of Text*. pp. 115-131. Cambridge, MA: MIT Press.

Smith, J. B. & Weiss, S. F. (1988) "Hypertext". *Communications of the ACM*. 31(7), 816-819.

Steinberg, D. (1990, September). "Multimedia is the Message". *PC Computing*, pp. 135-150.

Stevens, S. M., (1989) "Intelligent Interactive Video Simulation of a Code, Inspection" *Communications of the ACM*. 32(7), 832-843.

Tinker, M. (1989). "DVI Parallel Image Compression", *Communications of the ACM*. 32(7), 844-851.

Van Dam, A. (1988). "Hypertext '87 Keynote Address", *Communications of the ACM*. 31(7), 887-895.

Yankelovich, N., & Meyrowitz, N. (1985, October). "Reading and Writing the Electronic Book", *IEEE Computer*. pp. 15-30.

Younggren, G. (1987). "Using an Object-Oriented Programming Language to Create Audience-Driven Hypermedia Environments", Apple Computer, Inc.